This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

Shifu2: A Network Representation Learning Based Model for Advisor-advisee Relationship Mining

Jiaying Liu, Feng Xia, *Senior Member, IEEE,* Lei Wang, Bo Xu, Xiangjie Kong, *Senior Member, IEEE,* Hanghang Tong, *Senior Member, IEEE,* and Irwin King, *Fellow, IEEE*

Abstract—The advisor-advisee relationship represents direct knowledge heritage, and such relationship may not be readily available from academic libraries and search engines. This work aims to discover advisor-advisee relationships hidden behind scientific collaboration networks. For this purpose, we propose a novel model based on Network Representation Learning (NRL), namely Shifu2, which takes the collaboration network as input and the identified advisor-advisee relationship as output. In contrast to existing NRL models, Shifu2 considers not only the network structure but also the semantic information of nodes and edges. Shifu2 encodes nodes and edges into low-dimensional vectors respectively, both of which are then utilized to identify advisor-advisee relationships. Experimental results illustrate improved stability and effectiveness of the proposed model over state-of-the-art methods. In addition, we generate a large-scale academic genealogy dataset by taking advantage of Shifu2.

Index Terms—Social network analysis, Relation extraction, Network representation learning, Scientific collaboration network, Advisor-advisee relationship.

1 Introduction

It is well-known that academic network can be formed according different types of relationships, such as colleagues, friends, and advisor-advisee relationships. These relationships usually reflect different interpersonal interactions. For example, in advisor-advisee relationships, a PhD candidate's research topic is usually determined by his/her advisor (i.e., supervisor). While in friendships, one's daily schedule may be reflected by his/her friends. These interactions govern the dynamics and the complexity of social networks. To better model the interaction based on Network Science, a concrete network is abstracted into a graph consisting of nodes and edges, where nodes represent the entities and the edges indicate the different relationships. Hence, we can model the influence of nodes and edges from both local and global perspectives using graph theoretical methods and machine learning techniques.

With the rapid growth of scholarly information and artificial intelligence [1], researchers have shown rapidly-increasing interest in exploring big scholarly data (BSD) [2]. BSD contains not only scholarly records including papers, authors, venues, and citations, but also other related data such as scientific networks and digital libraries. Obviously, the data can form various types of networks within the science of science [3]. For example, in an academic collaboration network, nodes usually represent scholars and edges mean that connected scholars have even collaborated with each other. Another important type, the citation network, contains a large number of papers (represented as nodes) and citation relationships (represented as edges). Based on the analysis of these networks, we can effectively retrieve useful information hidden behind the nodes. Newman et al. [4] analyze the scientific collaboration networks and highlight the differences in collaboration patterns between different research fields. Wang et al. [5] predict scientific collaboration sustainability from the perspectives of collaboration times and collaboration duration. Yu et al. [6] recognize academic collaborative teams by exploring collaboration intensity.

As a relationship in scientific collaboration networks, the advisor-advisee relationship is vitally important for scholars. Generally speaking, each newcomer in academia will be advised/supervised by an advisor. The hypothesis that both advisees and advisors benefit from advisor-advisee relationships is proved by copious literature [7], [8]. Specifically, from the advisees' perspective, they receive guidance and support (i.e., funds for scientific research, career advice). Malmgren et al. [9] also point out that the advisor-advisee relationship is important to academic organizations because advisees usually would like to be committed to their organizations after graduation. Therefore, identifying advisor-advisee relationships will benefit many significant applications, such as double-blind peer review, scientific genealogy generation, and scientific career analysis. More importantly, mining and analyzing of advisor-advisee relationships will help us better understand underlying principles of the academic society from the perspectives of collaborative teamwork formation [10], [11] and scientific entities modeling [12], [13], [14].

- *J. Liu, F. Xia, L. Wang, B. Xu, and X. Kong are with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China.*
- *H. Tong is with School of Computing, Informatics and Decision Systems Engineering, Arizona State University, USA.*
- *I. King is with Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.*

Corresponding author: Feng Xia; email: f.xia@ieee.org.

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

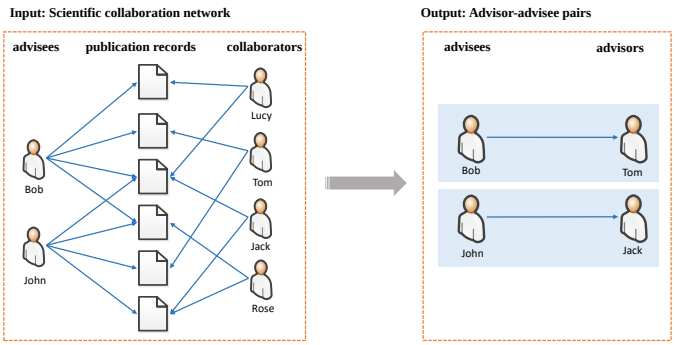


Fig. 1. An example of advisee-advisor relationship mining in the scientific collaboration network.

In practice, the lack of high-quality data about academic mentorships significantly limits the exploration of advisor-advisee relationships. There are some projects that aim at helping users track the academic genealogy, such as the Mathematics Genealogy Project[1], Neurotree[2], and the MPACT Project[3]. But all of projects rely on manual processing, which limits the number of records. Meanwhile, few of them provide scientific related information for advisor-advisee pairs, such as publication records and collaboration information. Although some researchers [15], [16], [17] have proposed the methods to deduce mentorships from academic publication networks, they ignore dynamics and complexity in science itself. It is still an open issue to develop appropriate methods that can extract advisor-advisee relationships automatically from scientific digital libraries. However, it suffers from a number of challenges due to the inherent properties and environmental attributes of the relationship. For instance, the relationship is time-dependent. The identity of an advisee will change over time. How to control this time-dependent factor is critical to the solution.

To tackle the above challenges, in this paper, we propose an effective model named Shifu2, which takes advantage of Network Representation Learning (NRL) techniques to form an intelligent solution for mining advisor-advisee relationships hidden in a scientific collaboration network. To better elaborate the problem, Fig. 1 presents an example of the advisor-advisee relationship identification in the network. Shifu2 is an NRL model. Specifically, we regard the advisor-advisee relationship mining problem as a classification problem. The collaborators and the attributes of each scholar are modeled as joint vectors. We design an efficient algorithm to optimize the process of vector representation composed of edge attributes and node attributes. We conduct extensive experiments to evaluate the effectiveness of our model. The results illustrate that the best performance of this method can achieve the accuracy of 92%, leading by at least 5% against the baseline methods.

Furthermore, we analyze the differences between advisor-advisee relationships over varying intervals of time. In the application part, we consider these differences and add an age adjustment parameter and a disciplinary adjustment parameter. Thus, it can be applied to the entire dataset which contains multi-disciplinary publication records through a long history.

In summary, the main contributions of this paper can be summarized as follows:

- **A Novel Mining Model**. We devise Shifu2 based on the NRL technique, to identify advisor-advisee relationships hidden in the scientific collaboration network. Unlike existing research, we consider semantic information of both nodes and edges for embedding. Experimental results demonstrate outstanding capabilities of Shifu2 for the task.
- **New Knowledge.** We discover the differences between each discipline with respect to the structures of advisor-advisee collaboration networks. Based on the obtained observations, we add the parameters of time adjustment and disciplinary adjustment to make Shifu2 universal. Consequently it can eliminate temporal/chronological differences as well as disciplinary differences.
- **A Benchmark Dataset**. By applying Shifu2 onto the entire pre-processed dataset (i.e., Microsoft Academic Graph (MAG)[4]), we generate a large-scale dataset containing not only advisor-advisee pairs but also the academic attributes and publication records of each scholar.

The rest of the paper is organized as follows. Section 2 lays out the research scope and introduces related work. Section 3 formally formulates the problem and presents the overall architecture of the proposed solution. Section 4 describes experimental settings and presents the results to illustrate the effectiveness of Shifu2. Besides, we also analyze the results and present the findings focusing on the application and visualization. Finally, Section 5 concludes the paper.

2 Related Work

The related work is divided into three parts. The first part deals with relation extraction (RE) in social networks. The second part reviews existing work for the advisor-advisee relationship identification. The third part focuses on NRL techniques.

2.1 Social Relation Extraction

RE is an important topic in knowledge graphs, which focuses on extracting relationships among entities to enrich existing information. In modeling relationships in knowledge graphs, relationships between entities are usually described as "head" + "relation" = "tail", i.e., "advisors" + "advising" = "advisees". Social relationship extraction (SRE) is an important subtask of RE, which focuses on mining social interrelationships between entities. Generally, the social network is defined as $G = (V, E)$, where V means objectives and $E \subseteq (V \times V)$ are edges between them. The edges E consists of two types, labeled E_L and unlabeled E_U. Then

1. https://www.genealogy.math.ndsu.nodak.edu/
2. https://neurotree.org/neurotree/
3. http://www.ibiblio.org/mpact/
4. https://www.openacademic.ai/oag/

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

the problem of SRE can be described as: How to predict the labels over each edge in E_U based on G and E_L?

The most commonly used techniques in SRE can be classified into four categories including similarity measures, statistical relational learning measures, graph mining measures, and machine learning techniques. Previous studies in SRE mainly use these techniques for unstructured data (e.g., text data, web pages, and the corpus of literature). Up to now, a number of studies have been carried out to identify intimate relationships. Diehl et al. [18] utilize a supervised ranking approach to identify the manager-subordinate relationship. Reviewing the development of SRE techniques, the major problems are:

(1) Relationships in real-world networks are complex and dynamic, which means that entities will play different roles under different scenarios. The interactions between entities also change over time. A single label can not describe vertices well because it can provide neither sufficient descriptions nor dynamic characteristics.

(2) In comparison with RE in knowledge graphs where relations are well pre-defined with human efforts, relations between vertices in social networks are usually invisible. We should consider how to obtain invisible information with the help of existing techniques.

2.2 Advisor-advisee Relationship Identification

The advisor-advisee relationship is one of the most important relationships in scientific collaboration networks. It can benefit many academic related applications such as advisor recommendation and reviewer recommendation. However, identification of such relationship is not straight forwards and the study still remains a challenge. Wang et al. [16] propose TPFG to extract the advisor-advisee relationship based on the probabilistic factor graph. Zhao et al. [17] capitalize on a deep model equipped with improved Refresh Gate Recurrent Units to identify the relationship. Another framework to identify the advisor-advisee relationship proposed by Wang et al. [15] exploits a deep learning method.

Shifu2 is built on top of a previous work named Shifu [15]. In the process of determining edge and node attributes, we refer to the feature selection part of Shifu. However, compared with Shifu, Shifu2 has further improvement in model selection, data processing, and application perspectives. Specifically, the main differences between Shifu2 and Shifu are:

(1) Shifu utilizes the Digital Bibliography & Library Project (DBLP)[5] dataset, which only contains publication records in the field of Computer Science. It means that the task is limited to the field of Computer Science. In Shifu2, we use the MAG dataset containing paper information for more fields. We crawl the real advisor-advisee pairs in six major fields including Chemistry, Computer Science, Economics, Engineering, Mathematics, and Physics. Through individual training of data in different fields, Shifu2 model has better applicability and scalability.

(2) Shifu utilizes supervised learning using a deep learning based model. Compared with Shifu, Shifu2 is an NRL model and represents each node in a low-dimensional space first. Then it capitalizes on the representations as the input of the task. As a result, Shifu2 is able to apply in large-scale networks and archive good performance in a short time. Shifu2 can better promote the convergence and reasoning, especially when labeled data is sparse. In addition, Shifu2 integrates attributes into node and edge attributes, which can better verify the effect of different attributes on the model.

(3) In the application part, the authors directly apply Shifu model on the large-scale network. On the contrary, Shifu2 preprocesses author name disambiguation, disciplinary differences elimination, and re-scaled number of publications to solve the problems of author name duplication, disciplinary differences, and temporal effect, respectively.

5. https://dblp.uni-trier.de/

2.3 Network Representation Learning

With the development of machine learning techniques, how to represent nodes information in various networks has become a critical issue. NRL is becoming a key instrument in modeling network structure. It learns the representations for nodes and adopts them as features for downstream applications such as link prediction and node classification. Formally, NRL learns a real vector $R_v \in R^k$ for each node, where the dimension of k is much smaller than the total number of nodes $|V|$. The learning process can either be unsupervised or semi-supervised.

Existing NRL models can be broadly divided into two categories: one is based on network structures and the other considers external information such as text information and label categories. DeepWalk [19] and LINE [20] are typical models which attempt to learn the representations based on local network structures. To preserve the global network structure, Cao et al. [21] present GraRep to learn representations in weighted graphs. Wang et al. [22] propose M-NMF to preserve both the microscopic and community structure.

Recently, research on incorporating heterogeneous information into NRL models has received increasing interests [23], [24], [25], [26]. Among them, specific text information (e.g., labelling information) is usually used as edges' external information for node representation. Yang et al. [23] introduce TADW, which incorporates text information into NRL models based on matrix factorization. CANE [25] learns the context-aware embeddings for vertices based on the attention mechanism to model the semantic relations between themselves.

3 Design of Shifu2

In this paper, we focus on mining advisor-advisee relationships hidden in the scientific collaboration network. Although NRL techniques can effectively learn the complex structures of networks, it is easy to make the representations sub-optimal because shallow network embedding models cannot capture the highly nonlinear network structure. Taking advantage of deep learning models which have great ability to learn the complex network structure [27], [28], we adopt deep learning models to learn the representations for the target network.

Unlike existing studies, we incorporate attributes for both edges and nodes into NRL models and formalize the task as modeling the relations between nodes. In an

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

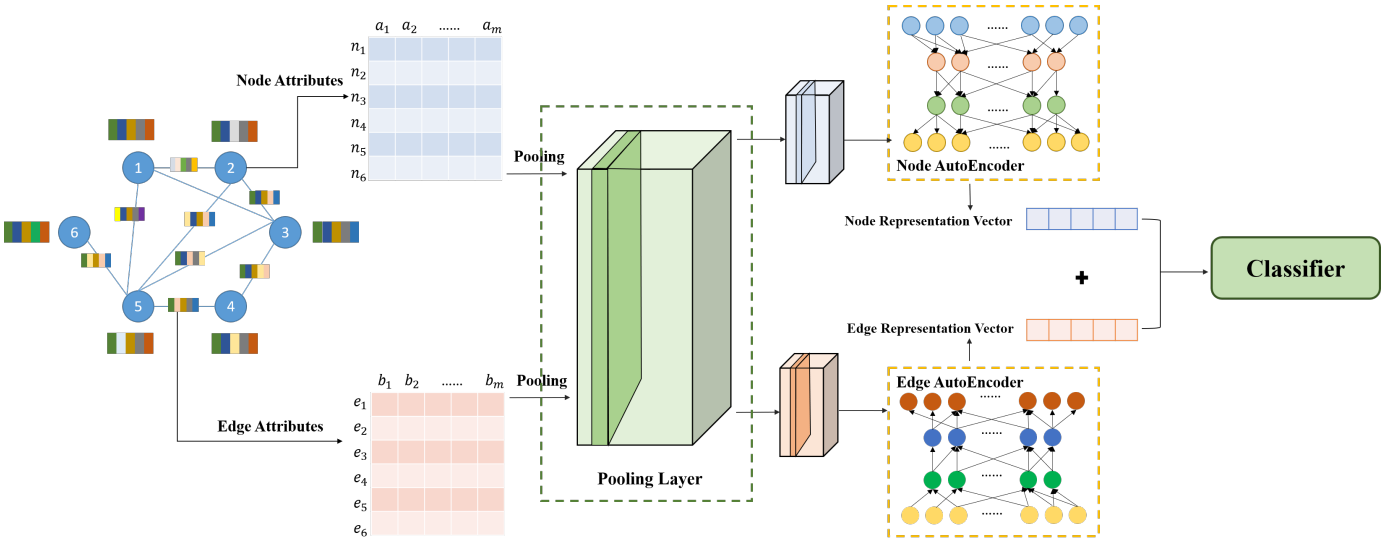


Fig. 2. Framework of Shifu2, which contains four key components: (1) edge representation construction, (2) node representation construction, (3) pooling, and (4) classification.

academic collaboration network, both nodes and edges attributes can reflect network properties. For example, from the perspective of node attributes (i.e., academic age), advisors often have a longer career than their advisees. At the same time, edge attributes such as collaboration times can reflect collaboration intensity. In order to preserve network structures in the joint space composed of node attributes and edge attributes, we exploit the proximity jointly by using a deep learning model. As shown in Fig. 2, Shifu2 contains four critical components, i.e., edge representation construction part, node representation construction part, pooling part, and classification part. The node autoencoder is used to capture the local network structure and the edge autoencoder is used to preserve the collaboration information.

To clearly illustrate the problem, we first define the notations used in Shifu2 framework. Then we present the details of how to construct the representations for nodes and edges, respectively. At last, we introduce the overall reconstruction mechanism of Shifu2.

3.1 Problem Formulation

Since we focus on the collaboration in advisor-advisee relationships, we select all papers published by target scholars to construct the collaboration network $G = (V, E)$, where the node $v_i \in V$ represents the scholar i and edges in E weighted by collaboration times represent co-author relationships. Here we suppose that the advisor of i is one of i's collaborators, and we use $v_j \in C_i$ to represent his/her collaborator, where C_i is the set of i's collaborators. Shifu2 considers external information of edges and nodes at the same time.

Definition 1. **Node Attributes**. Node attributes represent the inherent attributes of nodes. In the proposed model, it can be regarded as the local properties of scholars, such as the institution, the academic age, the number of publications, etc. We use node attributes to reflect the scholars' academic performance and the similarity of two scholars. For example, an advisor and his/her advisee are more likely to belong to the same institution, while the advisor may publish more papers than his/her students. $\boldsymbol{A_n} \in \mathbb{R}^{n_1 \times m_1}$ preserves the node attributes, where m_1 is node attribute categories and $n_1 = |V|$ is the total number of nodes in the network.

Definition 2. **Edge Attributes**. Edge attributes aim to depict relationship intensity or other properties. In Shifu2, they are used to describe the collaboration intensity between scholars, such as collaboration times, collaboration duration and so on. In contrast to other collaborators, the collaboration pattern between advisees and advisors are special. For example, advisees tend to collaborate frequently with their advisors at the early stage of their careers. Similarly, we use $\boldsymbol{A_e} \in \mathbb{R}^{n_2 \times m_2}$ to preserve the edge attributes, where $n_2 = |E|$ is the number of edges and edge attribute categories is defined as m_2.

TABLE 1 lists other symbols used in the representation learning process.

TABLE 1
Description of notations

Notation	Description
$n_1 = \|V\|$	the number of nodes in the collaboration network G
$n_2 = \|E\|$	the number of edges in the collaboration network G
L	the label set for edges
m_1	the number of node attribute categories
m_2	the number of edge attribute categories
d_1	the dimension of node representation
d_2	the dimension of edge representation
r	the learning rate for the autoencoder
$\boldsymbol{A_n} \in \mathbb{R}^{n_1 \times m_1}$	the node attribute information matrix
$\boldsymbol{A_e} \in \mathbb{R}^{n_2 \times m_2}$	the edge attribute information matrix
$\boldsymbol{H} \in \mathbb{R}^{n_1 \times d}$	the final representation of the network

Based on the notations explained above, the problem of advisor-advisee relationship mining can be described as follows:

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

Input: A scholar i who is associated with $\boldsymbol{A_n}$ and $\boldsymbol{A_e}$, the collaboration network G, and C_i.

Output: Who is i's advisor?

3.2 Framework of Shifu2

We assume both nodes and edges attributes can preserve the interactions between advisors and advisees in a collaboration network. From the perspective of node attributes, advisors usually publish more papers than their advisees. Meanwhile, advisors will have a relatively long academic career, which leads to cumulative advantages, both in academic performance and resources. On the other hand, for edges attributes, the collaboration mechanism between advisors and advisees is obviously different from others. From the perspective of advisees, they collaborate more closely with their advisors. So we need to model the attribute proximity for both nodes and edges.

With this hypothesis, the objective of the collaboration network re-construction part in Shifu2 is to minimize the construction loss as:

$$L_{sf} = L_a + L_e \tag{1}$$

where L_a and L_e represent the reconstructed loss for the processes of node representation learning and edge representation learning. Followings give details about each part.

3.2.1 *Node Representation Learning*

Nodes attributes are highly correlated with the network structure in social networks [29]. In order to preserve the proximity of each scholar from the perspective of nodes connection, motivated by the method proposed by Tu et al. [26], we adopt a deep autoencoder to get the embedding representation $\boldsymbol{A'}$. The hidden layers of the autoencoder are considered as two parts: the encoder part and the decoder part. The layers consistently encode and decode the input data. The output of the $(i-1)th$ layer is considered as the input of the ith layer, which can ensure that the output is equal to the input. The hidden layers can automatically capture the characteristics of input data and keep them unchanged.

For each scholar and his/her collaborators in the collaboration network, we use the adjacency matrix represented by $\boldsymbol{A}$ as the input of the node autoencoder, where $\boldsymbol{A}_{ij} = 1$ if $(v_i, v_j) \in E$ and $\boldsymbol{A}_{ij} = 0$, otherwise. Meanwhile, we consider the attributes listed in TABLE 2 for each node in the network.

TABLE 2
Description of node features

Notation	Description
aa_i	the academic age of the scholar i
aa_j	the academic age of the scholar j
org_i	the organization of the scholar i
org_j	the organization of the scholar j
np_i	the number of publications of i before collaborating with j
np_j	the number of publications of j before collaborating with i

aa represents a scholar's academic age when he/she began to collaborate with his/her collaborators. It can be calculated as (2):

$$aa = y_c - y_f \tag{2}$$

where y_f is the year when i published the first paper and y_c is the year when i first co-authored with his/her collaborators. For example, if scholar i published his/her first paper in 1987 and the first co-authored paper with j is published in 2000, then his aa when he first collaborated with j is 13 (2000-1987 = 13).

In each hidden layer, we adopt the following no-linear transformation function:

$$\begin{aligned} \boldsymbol{h}^a_{(1)} &= f_a(\boldsymbol{W}^a_{(1)}\boldsymbol{x} + \boldsymbol{b}^a_{(1)}) \\ \boldsymbol{h}^a_{(i)} &= f_a(\boldsymbol{W}^a_{(i)}\boldsymbol{h}^a_{(i-1)} + \boldsymbol{b}^a_{(i)}), i = 2, ..., k \end{aligned} \tag{3}$$

where f_a is the activation function and $\boldsymbol{W}^a_{(i)}$, $\boldsymbol{b}^a_{(i)}$ represent the transformation matrix and the bias vector, respectively. k is the total number of hidden layers. Besides, we utilize the Sigmoid function to map vectors of arbitrary real values to the range $[0, 1]$.

The goal of the node representation part is to minimize the reconstruction error between the representation vectors and the original input. However, the number of non-zero elements in the input is far less than that of zero elements. Therefore, it is easy to reconstruct zero elements. To address this problem, we impose more penalties on reconstruction error of non-zero elements. According to Wang et al. [30], we add a non-zero penalty matrix $\boldsymbol{A''}$ and define the objective function as:

$$L_a = ||((\boldsymbol{A_n} \parallel \boldsymbol{A}) - \boldsymbol{A'}) \odot \boldsymbol{A''}||^2_F \tag{4}$$

where $\boldsymbol{A_n} \parallel \boldsymbol{A}$ is the concatenation operation of $\boldsymbol{A_n}$ and $\boldsymbol{A}$. Specifically, the concatenation operation $\parallel$ in this equation is to connect two matrices in the horizontal axis. Here we use a toy example to explain this operation clearly. Suppose that there are three nodes n_1, n_2, n_3 in the network G nd the adjacency matrix $\boldsymbol{A} = [[0,1,0],[1,0,1],[0,1,0]]$. Each node in G is associated with two attributes and the node attributes matrix is $\boldsymbol{A_n} = [[2,1],[1,0],[1,1]]$. Then $\boldsymbol{A_n} \parallel \boldsymbol{A} = [[0,1,0,2,1],[1,0,1,1,0],[0,1,0,1,1]]$. The concatenation operation is implemented by the function $numpy.concatenate((), axis = 1)$ in Python. If $\boldsymbol{A}_{ij} > 0$, then $\boldsymbol{A''}_{ij} = \rho, \rho > 1$, else $\boldsymbol{A''}_{ij} = 1$. By using the loss function as (4), the nodes with similar characteristics will be close in the embedding space.

We adopt Adaptive moment estimation (Adam) [31] method for first-order gradient-based optimization. For each step in the iteration process, we need to compute the gradients g_t at time t. Then we compute the first moment estimate m_t and the second raw moment estimate v_t, where m_t is the mean of the gradient g and v_t is the non-central variance of g_t. According to the bias-corrected $\widehat{m}_t$ and $\widehat{v}_t$, we can update the parameter θ_t. In summary, the optimization

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

process can be described as:

$$\begin{aligned} g_t &\longleftarrow \bigtriangledown_\theta f_t(\theta_{t-1}) \\ m_t &\longleftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \\ v_t &\longleftarrow \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2 \\ \widehat{m}_t &\longleftarrow m_t/(1-\beta_1^t) \\ \widehat{v}_t &\longleftarrow v_t/(1-\beta_2^t) \\ \theta_t &\longleftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t/(\sqrt{\widehat{v}_t}+\epsilon). \end{aligned} \tag{5}$$

Based on experience and experimental results, in our optimization process, we set 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ϵ.

3.2.2 Edge Representation Learning

In the edge representation construction process, we also employ the deep autoencoder to convert the attribute matrix to the low-dimensional vector representations. The reconstruction process and the implementation details are presented below.

In order to preserve the edge proximity, we take the edge attribute matrix as the input of edge autoencoder. We consider attribute information listed in TABLE 3 for each edge in the network.

TABLE 3
Description of edge features

Notation	Description
ad_{ij}	difference of academic age between i and j
ct_{ij}	the collaboration times of i and j
cd_{ij}	the collaboration duration of i and j
ft	the number of times i and j being the first two authors
lf	the number of times i and j being the first and the last authors
$kulc_{ij}^t$	the collaboration similarity between i and j

$kulc_y^{ij}$ is the collaboration similarity between scholar i and his/her collaborator j after t years since they first co-authored a paper [32]. It aims to depict the dynamics of collaboration. We compute $kulc$ as:

$$kulc_{ij}^t = \frac{np_{ij}}{2}(\frac{1}{np_i} + \frac{1}{np_j}) \tag{6}$$

where np_{ij} is the number of co-authored papers in t years, and np_i, np_j represent the number of publications of i, j, respectively.

The autoencoder adopts the edge attribute matrix as input. It encodes and decodes the vector in each no-linear transformation layers as:

$$\begin{aligned} \boldsymbol{h}_{(1)}^e &= f_e(\boldsymbol{W}_{(1)}^e a + \boldsymbol{b}_{(1)}^e) \\ \boldsymbol{h}_{(j)}^e &= f_e(\boldsymbol{W}_{(j)}^e \boldsymbol{h}_{(j-1)}^e + \boldsymbol{b}_{(j)}^e), j = 2, ..., m \end{aligned} \tag{7}$$

f_e is the activation function and m is the number of hidden layers of the encoder. In the jth layer, we use $\boldsymbol{W}_{(j)}^e$ and $\boldsymbol{b}_{(j)}^e$ to represent the transformation matrix and the bias vector, respectively. Specifically, the input of the edge autoencoder is dense, so we utilize the dropout [33] to overcome the overfitting. Similar to the process of nodes representation, Sigmoid function is employed as the activation function to get the edge representation matrix $\boldsymbol{E}'$. The reconstructed loss is computed as:

$$L_e = ||\boldsymbol{A_e} - \boldsymbol{E}'||_F^2. \tag{8}$$

Hence, we can minimize the distances between the reconstructed representations and the original input.

3.2.3 Pooling Layer

Some fields contain thousands of scholars. It is difficult to ensure the computational time and memory for the running control with numerous vectors. As a result, we add a pooling layer to compress input features. We first reduce each adjacency vector to 1000 dimensions and calculate the mean of reduced vectors accordingly as the input of encoders.

3.2.4 Advisor-advisee Relationship Identification

After the processes of nodes and edges reconstruction, we can obtain the low-dimensional representations of the collaboration network. Then we need to use a supervised classifier to predict the label for each edge.

More specifically, for the unlabeled edge $e = (i, j)$, Shifu2 aims to identify the label for e. To achieve this goal, we add a supervised classifier in our Shifu2 model, which takes the output of the last hidden layer as input and returns the classification results. We adopt logistic regression [34] as the classification method in our model. Mathematically, the loss function L_{lr} is computed as:

$$L_{lr} = |L - (\boldsymbol{D} \cdot \boldsymbol{W}^l + \boldsymbol{B}^l)| \tag{9}$$

where $\boldsymbol{D}$ is the concatenation of two encoders. $\boldsymbol{W}^l$ is the weight of logistic regression and $\boldsymbol{B}^l$ is the bias. In this part, we also use Adam method for stochastic optimization.

3.2.5 Overall Reconstruction

After the representation process from the perspectives of both nodes and edges, in order to preserve nodes and edges properties at the same time, we calculate the joint loss function to optimize the objective as:

$$\begin{aligned} L_{sum} =& L_{sf} + \beta L_{lr} + \\ & \alpha(\Sigma_i(\boldsymbol{W}_i^e + \boldsymbol{B}_i^e) + \gamma\Sigma_j(\boldsymbol{W}_j^a + \boldsymbol{B}_j^a)) \end{aligned} \tag{10}$$

where α and β are two hyperparameters to adjust the weights of the edge encoder and the node encoder.

Since the edge encoder is a process of dimension rise and the node encoder is a process of dimension reduction, the sum of weight and bias of the node encoder is much larger than the edge side. We use the regularization parameter γ to balance in the middle, to prevent the value of the node side overweight the edge side. In the experiment, we set γ as 0.01. If we combine (10) with (1), (4), (8) and (9), then the overall loss function can be rewritten as:

$$\begin{aligned} L_{sum} =& ||((\boldsymbol{A_n} \parallel \boldsymbol{A}) - \boldsymbol{A'}) \odot \boldsymbol{A''}||_F^2 + ||\boldsymbol{A_e} - \boldsymbol{E'}||_F^2 + \\ & \beta|L - (\boldsymbol{D} \cdot \boldsymbol{W}^l + \boldsymbol{B}^l)| + \\ & \alpha(\Sigma_i(\boldsymbol{W}_i^e + \boldsymbol{B}_i^e) + \gamma\Sigma_j(\boldsymbol{W}_j^a + \boldsymbol{B}_j^a)). \end{aligned} \tag{11}$$

The goal of the aforementioned model is to minimize the loss function L_{sum}. In this paper, we achieve this goal by using back propagation algorithm (BP) with stochastic

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

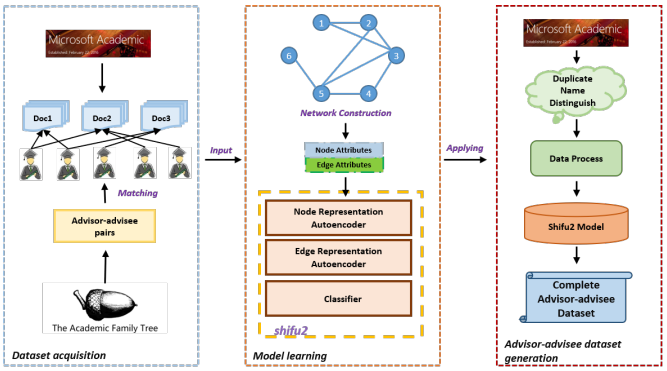


Fig. 3. Experiment procedures.

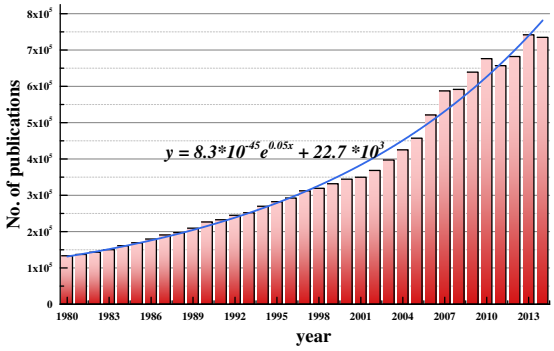


Fig. 4. Number of new records per year in the MAG dataset.

gradient descent, e.g., $W_{ji} = W_{ji} - \delta \frac{\partial}{\partial W_{ji}} L(X, Y)$. If $z = Wx + b$, the gradient can be defined as:

$$\begin{aligned} \frac{\partial L(X,Y)}{\partial W_{ji}} &= \Sigma_{i=1}^{n} \frac{\partial L(X,Y)}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial W_{ji}} = \Sigma_{i=1}^{n} \rho_j x_i^T \\ \frac{\partial L(X,Y)}{\partial b_j} &= \Sigma_{i=1}^{n} \frac{\partial L(X,Y)}{\partial Z_j} \cdot \frac{\partial Z_j}{\partial b_j} = \Sigma_{i=1}^{n} \rho_j \end{aligned} \tag{12}$$

where $\rho_j = \partial L(X,Y)/\partial z_j$ is the reconstruction error between the activation and the target. n represents the number of samples.

If $f'(.)$ represents the partial derivative of $f(.)$, then the updated error can be summarized as:

$$\begin{aligned} \rho_j^{(Y)} &= -\Sigma_{i=1}^{n}(y_{ij} - h_{ij}) \cdot f'(z_j^{(Y)}) \\ \rho_j^{(H)} &= \Sigma_{i=1}^{n} W_{ji}^{H} \rho_i^{(Y)} \cdot f'(z_j^{(H)}). \end{aligned} \tag{13}$$

The overall architecture of the algorithm is summarized in Algorithm 1.

Algorithm 1 The advisor-advisee relationship mining algorithm.

Input: $\boldsymbol{A}$, $\boldsymbol{A_n}$, $\boldsymbol{A_e}$, $\alpha, \beta, \gamma, r, L$, and the convergence condition ϵ.
Output: l for each $e = (i, j)$

Initiate: randomly initiate $\theta_1 \leftarrow \{\boldsymbol{W}_i^e, \boldsymbol{B}_i^e\}_{i=1}^m$, $\theta_2 \leftarrow \{\boldsymbol{W}_j^a, \boldsymbol{B}_j^a\}_{j=1}^n$, and $\theta_3 \leftarrow \{\boldsymbol{W}^l, \boldsymbol{B}^l\}$.
Pre-train: set $r = 0.01$, train edge auto-encoder and node auto-encoder independently.
Train:
while ϵ is true **do**
    randomly generate a batch of data from advisor-advisee edge data and advisors and advisee's attributes
    if $\boldsymbol{A}_{ij} > 0$ **then**
        $\boldsymbol{A}''_{ij} \leftarrow \rho$, $\rho > 1$
    else
        $\boldsymbol{A}''_{ij} \leftarrow 1$
    end if
    calculate L_a based on (4), get reconstructed $\boldsymbol{A'}$
    calculate L_e based on (8), get reconstructed $\boldsymbol{E'}$
    concatenate $\boldsymbol{D} \leftarrow [\boldsymbol{A'}, \boldsymbol{E'}]$
    calculate L_{sum} based on (11)
    update $\theta_1, \theta_2, \theta_3$
end while

3.3 Algorithm analysis

In the proposed model, we first generate $\boldsymbol{A'}$ and $\boldsymbol{E'}$ through the node autoencoder and the edge autoencoder. The complexity of generating the above embedding matrix is $O(c_e n_e d_e I_e + c_a n_a d_a I_a)$, where c_e, c_a are the size of samples. n_e, n_a are the dimension of the attributes. d_e, d_a are the maximum dimension of the hidden layer, and I_e, I_a are iterations times. For the logistic regression process, the computational complexity is $O(n * k + k)$, where n is the sample size and k represents the feature dimension, which is related to the embedding dimension of the encoders. It is not difficult to see that the total training complexity of Shifu2 is $O(c_e n_e d_e I_e + c_a n_a d_a I_a + nk + k)$. Approximately, the time complexity of Shifu2 is $O(cndI)$.

4 Experiments

This section presents the experimental details to assess the accuracy and the effectiveness of the proposed model. Fig. 3 describes the complete process of the task. Below we will give the details of each procedure.

4.1 Experimental Setup

4.1.1 *Datasets*

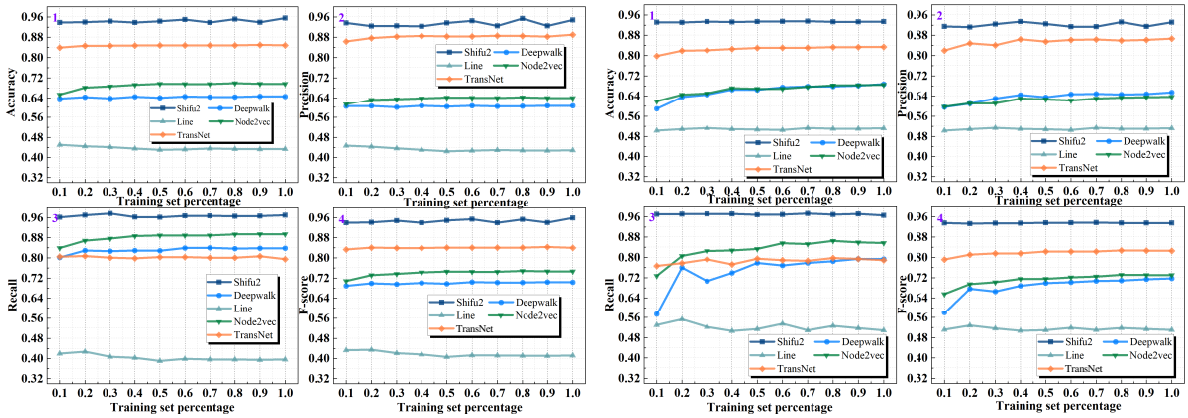
As mentioned previously, we use the MAG which is a widely used dataset containing scientific publication records to construct the collaboration network. In order to investigate the accuracy and the effectiveness of Shifu2, we need some ground truth advisor-advisee pairs. We extract the ground truth advisor-advisee pairs from The Academic Family Tree (AFT)[6], which is a user content-driven web database storing academic genealogy. We crawl the realistic advisor-advisee pairs in the fields of Chemistry, Computer Science, Economics, Engineering, Mathematics, and Physics. TABLE 4 lists the statistics for advisor-advisee pairs in major research fields.

Based on the ground truth advisor-advisee dataset, which only partially covers the authors in the MAG, we further randomly separate the dataset into two sub-datasets: the training set and the test set. The training set contains advisor-advisee pairs who co-authored the first paper during 2000-2006, and the others are used as the test set.

6. https://academictree.org/

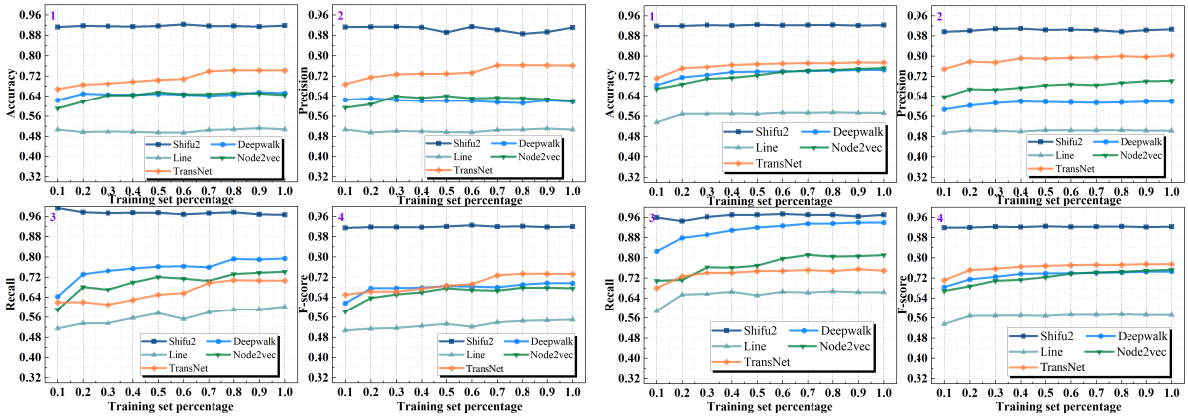
1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering



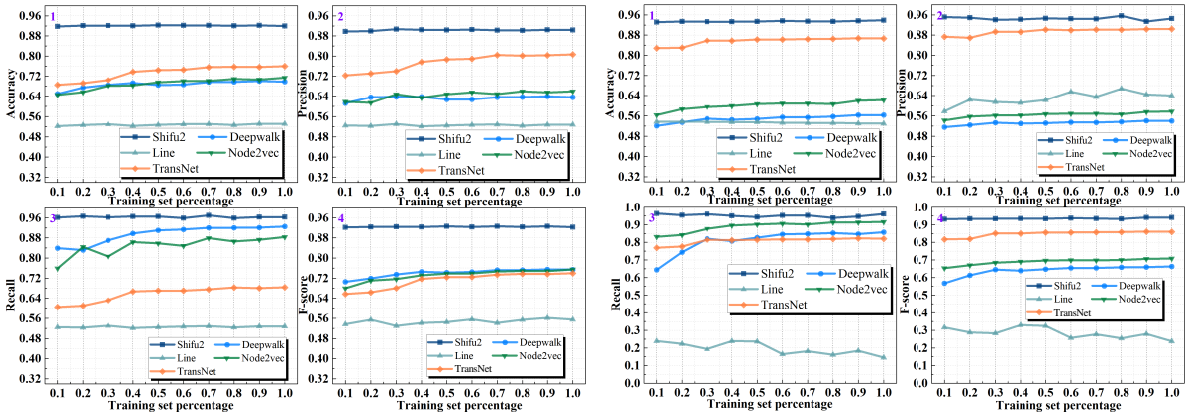
(a) Chemistry

(b) Computer Science



(c) Economics

(d) Engineering



(e) Mathematics

(f) Physics

Fig. 5. Performance of different methods with different sizes of training data.

TABLE 4
Statistics of advisor-advisee pairs in each field

Field	Number of advisor-advisee pairs	Time period
Chemistry	28,449	2000-2010
Computer Science	8853	2000-2010
Economics	2666	2000-2010
Engineering	9176	2000-2010
Mathematics	5962	2000-2010
Physics	7654	2000-2010
Total	62,760	

4.1.2 *Datasets Pre-processing*

Before applying Shifu2 onto the entire MAG dataset to generate the academic genealogy, there are a set of challenges that must be addressed:

- **Author name disambiguation**. The MAG only provides the publication related information such as title, authors, published year, and so on. Authors' names have not been pre-processed which means that if two scholars have the same name, their publications cannot be distinguished. Duplicated scholars'

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

personal information challenges the application of the model.

- **Disciplinary differences**. It is now well established from a variety of studies, that diverse collaborative ties exist in different research fields [35], [36]. The disciplinary differences may affect the performance of the model. It is a challenge that how to eliminate the disciplinary differences and make the model universal.
- **Temporal effect**. Fig. 4 illustrates that the number of publications in the MAG steadily grows over time, which may introduce influence on the law of advisor-advisee's collaboration. It is important to make sure that the results are not affected by the temporal effect.

In order to address these challenges, we pre-process the dataset as follows:

Author name disambiguation. To gain the complete and accurate publication records for each scholar, we need to infer their actual identities. Since the MAG dataset does not contain unique author identifiers, we conduct name disambiguation to overcome the problem of duplicate names. Accordingly to Sinatra et al [37], there are two main steps in the author name disambiguation: separation and mergence. The first step is separating all authors apart. It means that we should regard each author in the records as a unique one. The ultimate goal is to reduce duplicate identifies by merging authors iteratively. We consider the authors with identical names as the same individual if they meet one of the following criteria:

1) The two authors have been cited each other at least once;
2) The two authors have at least one co-author;
3) The two authors have at least one identical affiliation.

The name disambiguation process ends up with the condition that there are no author pairs to merge.

The matching process consists of two steps: the first step is to find the scholar in the MAG according to scholars' names obtained from AFT; the second step is to obtain features we need, such as publications and collaborators of these scholars. In the first step, we adopt a regular expression to present the name of each advisee and match them in the MAG dataset in their own fields. For example, if the scholars crawled from FTA major in computer science, then we match them in the field of computer science in the MAG dataset. We add all matching results in a dictionary as "the scholar's name in AFT: the scholar's name in the MAG", where "the scholar's name in the AFT" is the primary key of the dictionary. In addition, we establish a set to store all the information of the matched advisees. We match the advisor's name from matched advisee's collaborators in the same way. If we can match the advisor in the advisee's collaborators, we define these two collaborators as the ground truth advisor-advisee pair. We use this method to ensure that the matching of advisor-advisee pairs in the MAG dataset is what we really need. Finally, we can obtain the relevant information and digitalize it as the features for training.

Disciplinary differences elimination. As shown in TABLE 4, the training dataset contains six independent research fields. To observe the model performance, experiments are conducted independently in each field. When applying the model to the entire dataset, we add a disciplinary parameter δ_y^f for the field f in y (y represents the year), which is calculated as:

$$\delta_y^f = |F| * \frac{np_y^f}{\langle np \rangle_y} \tag{14}$$

where np_y^f is the total number of publications in y for f, and $\langle np \rangle_y$ is the average np calculated over all fields published in y. $|F|$ is the number of research categories. Since the papers in the MAG dataset are divided into 19 major disciplines, thus $|F| = 19$ in this paper.

Re-scaled number of publications. From Fig. 4, we observe that the growth of publications is in line with exponential distribution. The equation achieved by curve fitting is:

$$p = 8.3 * 10^{-45} e^{0.05y} + 22.7 * 10^3 \tag{15}$$

where p is the number of publication in year y. In order to compare the collaboration attributes between advisors and advisees starting their collaboration at the different time period, we use a re-scaled parameter, $\tilde{p}$, to gauge the publication records for each scholar:

$$\tilde{p} = p/\tau \tag{16}$$

where τ is the temporal correction factor, which is calculated as:

$$\tau = y' = 4.15 * 10^{-46} e^{0.05y}. \tag{17}$$

When calculating the collaboration features, we use the re-scaled dataset to prevent the model performance from temporal bias.

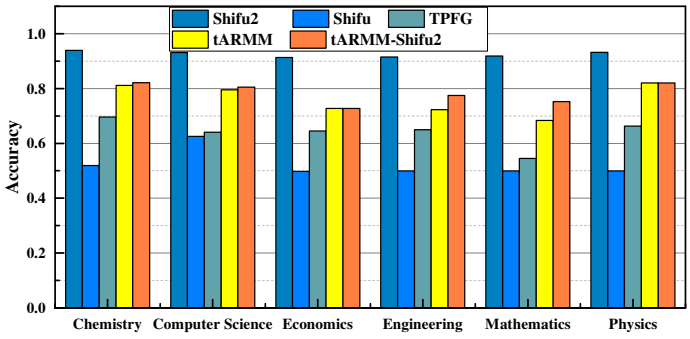


Fig. 6. Performance in terms of accuracy.

4.1.3 *Baselines*

We compare Shifu2 with the following baseline models:

- **DeepWalk** [19]. DeepWalk uses random walks to obtain the local information of the network and treats the walks as equivalent sentences. By employing skip-gram, it learns the latent representations for vertices.
- **LINE** [20]. LINE eliminates the limitation of classical stochastic gradient descent by using the edge-sampling algorithm. It can preserve both local and

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

TABLE 5
Advisor-advisee relationship identification performance of different methods

	Chemistry					Computer Science				
Metrics \ Method	Shifu2	Deepwalk	Line	Node2vec	TransNet	Shifu2	Deepwalk	Line	Node2vec	TransNet
Accuracy	**0.939**	0.646	0.451	0.699	0.850	**0.931**	0.681	0.514	0.685	0.834
Precision	**0.925**	0.606	0.447	0.643	0.890	**0.912**	0.655	0.514	0.637	0.867
Recall	**0.958**	0.840	0.427	0.894	0.808	**0.959**	0.794	0.552	0.865	0.797
F1-score	**0.941**	0.704	0.427	0.748	0.843	**0.933**	0.717	0.528	0.731	0.827
	Economics					Engineering				
Metrics \ Method	Shifu2	Deepwalk	Line	Node2vec	TransNet	Shifu2	Deepwalk	Line	Node2vec	TransNet
Accuracy	**0.913**	0.656	0.512	0.656	0.743	**0.915**	0.680	0.506	0.733	0.782
Precision	**0.877**	0.631	0.507	0.641	0.763	**0.889**	0.619	0.505	0.702	0.802
Recall	**0.961**	0.794	0.602	0.741	0.708	**0.952**	0.940	0.667	0.812	0.755
F1-score	**0.917**	0.697	0.550	0.678	0.734	**0.919**	0.746	0.575	0.753	0.891
	Mathematics					Physics				
Metrics \ Method	Shifu2	Deepwalk	Line	Node2vec	TransNet	Shifu2	Deepwalk	Line	Node2vec	TransNet
Accuracy	**0.919**	0.701	0.532	0.714	0.760	**0.932**	0.564	0.538	0.623	0.868
Precision	**0.898**	0.639	0.531	0.660	0.806	**0.935**	0.541	0.668	0.577	0.904
Recall	**0.947**	0.925	0.600	0.884	0.683	**0.959**	0.858	0.238	0.917	0.823
F1-score	**0.922**	0.755	0.537	0.755	0.739	**0.933**	0.663	0.329	0.708	0.861

global structures for following networks: direct/undirect networks and weighted/unweighted networks.

- **Node2vec** [38]. Node2vec is a semi-supervised method focusing on preserving neighbors' information for each node in the network. It can capture the diversity of connectivity patterns and learn low-dimensional feature representations for nodes effectively.
- **TransNet** [26]. TransNet is a knowledge graph-based framework, which translates the interactions between vertices to a translation operation. It considers the semantic information for each edge as a binary.

We also compare Shifu2 with existing solutions for identifying advisor-advisee relationships:

- **Shifu** [15]. Shifu is a deep learning model based on the stacked autoencoder. It takes scholars' personal properties and network characteristics as the input.
- **TPFG** [16]. TPFG is a time-constrained probabilistic model. It considers the task of advisor-advisee relationship as a jointly likelihood objective optimization problem.
- **tARMM** [17]. tARMM is a deep model based on the improved Refresh Gate Recurrent Units. It is inspired by the idea of variance Recurrent Neural Network models.
- **tARMM-Shifu2**. In tARMM-Shifu2, we modify the input of tARMM. We use node and edge attributes proposed in Shifu2 as the input of tARMM.

TABLE 6
Number of hidden units in each layer for node autoencoder and edge autoencoder

Encoder types	Group	No. layers	No. units
Node encoder	1	1	2000
	2	2	2000, 1000
	3	3	2000, 1000, 500
	4	4	2000, 1500, 1000, 500
	5	5	2000, 1500, 1000, 500, 300
Edge encoder	1	1	18
	2	2	18, 50
	3	3	18, 50, 70
	4	4	18, 30, 50, 70
	5	5	18, 30, 50, 70, 90

4.1.4 *Evaluation Metrics and Parameter Settings*

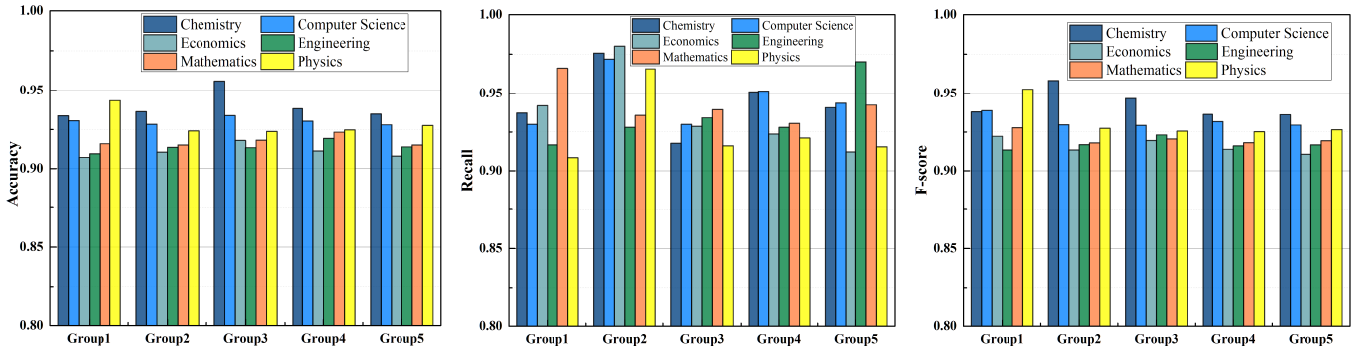
Since the task of advisor-advisee relationship mining can be regarded as a binary prediction problem, i.e., for each collaboration pair (i, j), this is the binary classification of whether j is i's advisor. To evaluate the performance of Shifu2, four widely used metrics for classification tasks are used in our experiments: Accuracy, Precision, Recall, and F1-score.

4.2 Results and Analysis

To evaluate the effectiveness of Shifu2 on identifying the advisor-advisee relationships, we conduct a series of ex-

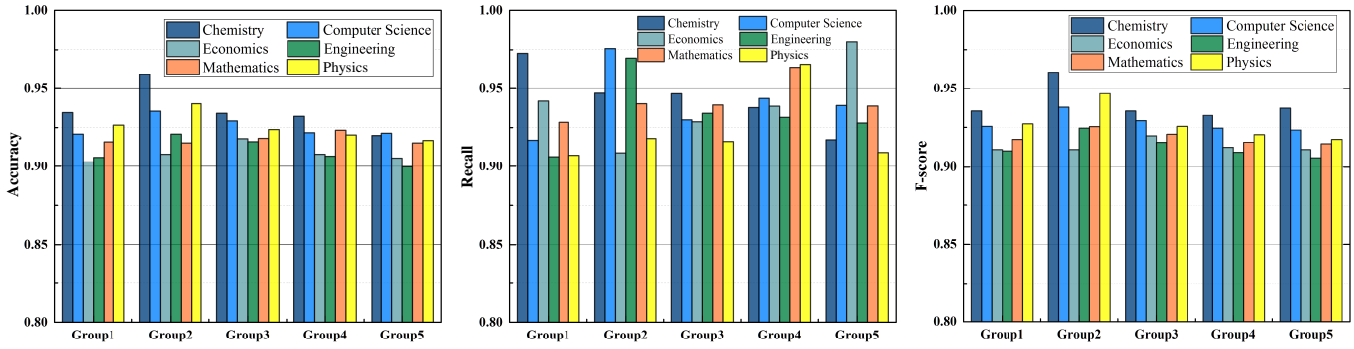
1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering



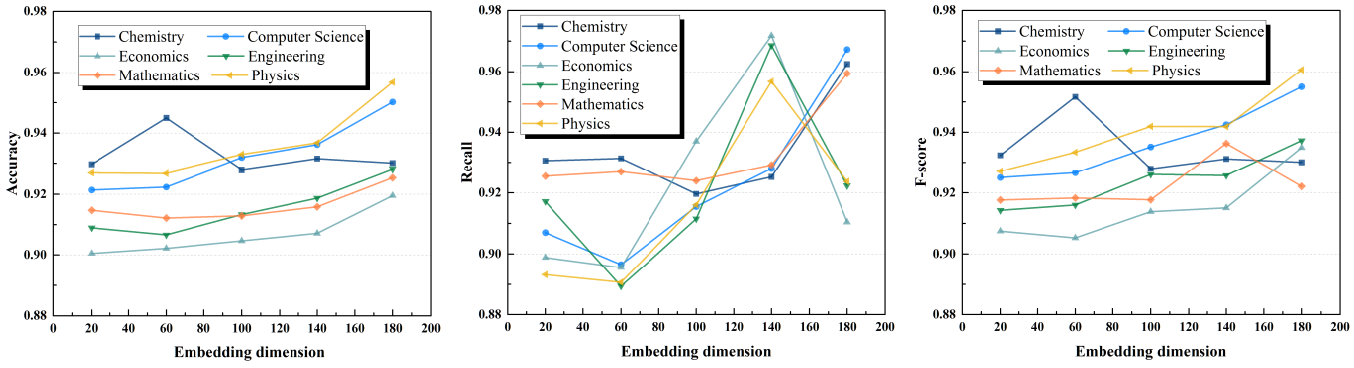
(a) Accuracy (b) Recall (c) F1-score

Fig. 7. Shifu2 performance with different hidden layers in node autoencoder.



(a) Accuracy (b) Recall (c) F1-score

Fig. 8. Shifu2 performance with different hidden layers in edge autoencoder.



(a) Accuracy (b) Recall (c) F1-score

Fig. 9. Shifu2 performance with respect to embedding dimension in node autoencoder and edge autoencoder.

periments from the following aspects. First, in order to illustrate how much Shifu2 can improve the embedding representation, we compare our proposed model with state-of-the-art NRL models mentioned in Section 4.1.3. Then we try to explore the effect of different assumptions in affecting the effectiveness of the model. Meanwhile, we compare the model performance between different research fields to find out the disciplinary differences.

4.2.1 *Effectiveness Evaluation*

TABLE 5 presents the results of comparing Shifu2 with all NRL based baselines. Note that, in the process of parameters setting in node2vec, p and q are selected in the set $\{0.25, 0.5, 1, 2, 4\}$. Finally, they are decided by the maximum of accuracy for each discipline. From the results, we can see that the task of advisor-advisee relationship identification achieves significant improvements by our proposed model. It demonstrates the effectiveness of Shifu2.

Besides, we vary the nodes for training from 10% to 100% of the training set. From Fig. 5 we can observe that Shifu2 consistently achieves stable performance in contrast to other baselines under different sizes of training data. Taking the accuracy rate as an example, Shifu2 is 30%, 38%, 28%, and 8% higher than deepwalk, line, node2vec, and Transnet in

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

TABLE 7
Performance of Shifu2 with different learning rates

	Chemistry				Computer Science				Economics			
Metrics \ Learning rate	0.001	0.005	0.01	0.1	0.001	0.005	0.01	0.1	0.001	0.005	0.01	0.1
Accuracy	0.936	0.943	**0.942**	0.932	0.911	0.921	**0.931**	0.929	0.892	0.905	**0.911**	0.911
Precision	0.943	0.945	0.937	0.919	0.928	0.929	0.926	0.905	0.880	0.888	0.886	0.880
Recall	0.932	0.945	0.949	0.948	0.896	0.916	0.941	0.958	0.920	0.934	0.949	0.9534
F1-score	0.938	0.945	0.943	0.933	0.911	0.923	0.933	0.931	0.898	0.910	0.916	0.915
	Engineering				Mathematics				Physics			
Metrics \ Learning rate	0.001	0.005	0.01	0.1	0.001	0.005	0.01	0.1	0.001	0.005	0.01	0.1
Accuracy	0.902	0.914	**0.916**	0.912	0.911	0.914	0.917	**0.919**	0.915	0.937	**0.943**	0.926
Precision	0.905	0.905	0.902	0.888	0.903	0.905	0.899	0.891	0.932	0.955	0.953	0.924
Recall	0.905	0.928	0.937	0.944	0.925	0.928	0.942	0.956	0.900	0.921	0.933	0.928
F1-score	0.905	0.916	0.919	0.915	0.914	0.916	0.920	0.922	0.916	0.938	0.943	0.926

the field of Chemistry, respectively.

Finally, we compare our model with existing methods for advisor-advisee relationships identification mentioned in Section 4.1.3. As shown in Fig. 6, Shifu2 achieves better performance consistently across all fields. Furthermore, we can observe that by using the node attributes and edges attributes proposed in Shifu2, the performance of the existing method such as tARMM has been improved.

4.2.2 *Parameter Sensitivity*

In this subsection, we study the effect of the following parameters: (1) number of hidden layers of the node autoencoder, (2) number of hidden layers of the edge autoencoder, (3) learning rate, (4) number of nodes used for training, (5) embedding dimension, and (6) input features.

Number of hidden layers. While the training set is determined, we should consider how to grid-search the number of hidden layers. Studies demonstrate that models will achieve better performance with more hidden layers. The highest accuracy can even reach 100%. However, models will be prone to over-fitting, thus the prediction performance will be sharply reduced on the test data. So we need to consider the effective depth of the training to make Shifu2 achieve better prediction performance without over-fitting. The number of units in each hidden layer needs to be determined accordingly.

In this paper, we need to determine the number of hidden layers for the node autoncoder and edge autoencoder, respectively. For the node autoencoder and the edge autoencoder, we choose the hidden layers from 1 to 5. The number of hidden units in each layer is presented in TABLE 6. Fig. 7 and Fig. 8 present the experimental results. Thus we can obtain the best architecture of our proposed model with three hidden layers for the node autoencoder, and two layers for the edge autoencoder.

Learning rate. Learning rate is a crucial parameter in Shifu2 because it controls the update speed of the model. If the learning rate is overlarge, the value of the loss function will move backwards and forwards around the minimum and will not converge. Otherwise, it will lead to a slow learning process. In TABLE 7, we discover that if the learning rate is set to 0.01, the model can achieve the best performance across almost all fields.

TABLE 9
Performance of Shifu2 without node autoencoder

	Chemistry		Computer Science	
Metrics \ Percentage	Shifu2	Shifu2-E	Shifu2	Shifu2-E
Accuracy	**0.939**	0.789	**0.931**	0.813
Precision	**0.925**	0.753	**0.912**	0.782
Recall	**0.958**	0.914	**0.959**	0.883
F1-score	**0.941**	0.823	**0.933**	0.830
	Economics		Engineering	
Metrics \ Percentage	Shifu2	Shifu2-E	Shifu2	Shifu2-E
Accuracy	**0.913**	0.507	**0.915**	0.736
Precision	**0.877**	0.506	**0.889**	0.718
Recall	**0.961**	0.602	**0.952**	0.873
F1-score	**0.917**	0.550	**0.919**	0.784
	Mathematics		Physics	
Metrics \ Percentage	Shifu2	Shifu2-E	Shifu2	Shifu2-E
Accuracy	**0.919**	0.702	**0.932**	0.846
Precision	**0.898**	0.670	**0.935**	0.827
Recall	**0.947**	0.889	**0.959**	0.890
F1-score	**0.922**	0.760	**0.933**	0.857

Number of nodes used for training. We set the number of nodes for training to be 20%, 40%, 60%, and 80% of the entire training data, respectively, to evaluate the sensitivity with respect to the size of training set. TABLE 8 presents the identification performance of the model. From the results, we observe that our model is well performed even though

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

TABLE 8
Performance of Shifu2 with different sizes of data

	Chemistry				Computer Science				Economics			
Metrics \ Percentage	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%
Accuracy	0.935	0.935	**0.939**	0.938	0.927	0.930	0.931	**0.931**	0.908	0.909	**0.913**	0.912
Precision	0.919	0.919	**0.925**	0.923	0.903	0.912	0.908	**0.912**	0.875	0.882	0.886	**0.877**
Recall	0.955	0.955	**0.958**	0.957	0.959	0.953	**0.959**	0.953	0.955	0.947	0.951	**0.961**
F1-score	0.937	0.936	**0.941**	0.940	0.930	0.932	**0.933**	0.932	0.913	0.913	**0.917**	0.917
	Engineering				Mathematics				Physics			
Metrics \ Percentage	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%
Accuracy	0.911	0.913	**0.915**	0.915	0.916	0.918	0.919	**0.919**	0.927	0.929	**0.932**	0.930
Precision	0.892	0.889	**0.889**	0.888	0.892	0.895	0.897	**0.898**	0.925	0.931	0.928	**0.935**
Recall	0.938	0.945	0.950	**0.952**	0.948	**0.950**	0.947	0.947	0.931	0.928	**0.937**	0.925
F1-score	0.914	0.916	**0.919**	0.919	0.919	0.921	0.921	**0.922**	0.928	0.929	**0.933**	0.930

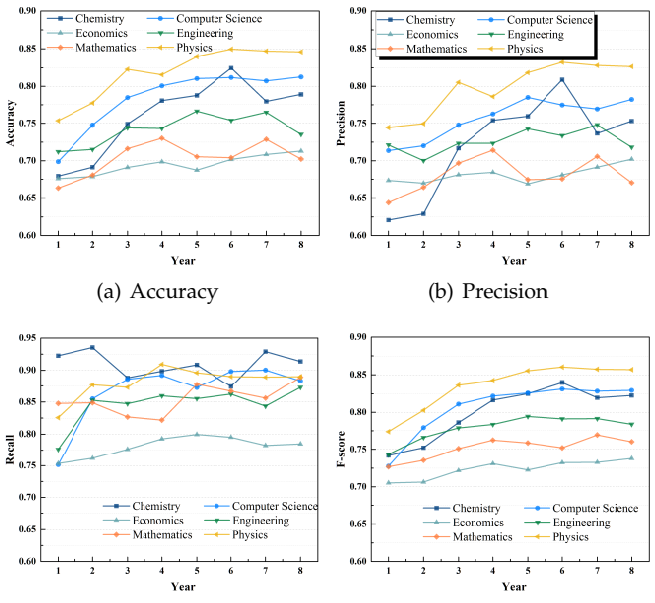
the training data is sparse. Shifu2 can reach the accuracy higher than 90% in all research fields with only 1000 nodes, which indicates the robustness of Shifu2.

Embedding dimension. After learning, we will get node representations and edge representations in terms of n-dimensional vectors, where n is the artificially set embedding dimension. To verify whether it influences the performance of the model, we vary it from 20 to 160 in Shifu2. The results are shown in Fig. 9. From the results we can see that, our model performs well with both low-dimensional representations and high-dimensional representations, which can be proved by the accuracy higher than 90% with different embedding dimensions. We also observe that the disciplinary differences do exist. For example, the performance of the model in terms of the accuracy keeps increasing with larger embedding dimensions for all research fields except chemistry. For chemistry, the accuracy first presents an increasing trend. After reaching a peak, it begins to decline.

Input features. A key ingredient of Shifu2 is the exploitation of node attributes and edge attributes. To validate the effectiveness of this mechanism, here we focus on examining the performance of Shifu2 without the node autoencoder (represented as Shifu2-E in TABLE 9) first. From TABLE 9, we can conclude that if we only use the edge autoencoder, it will achieve a relatively poor performance. The use of node attributes clearly improves the performances.

An advisee is usually supervised by an advisor for a specific period of time (i.e., not forever), during which they collaborate with each other closely. In many (if not most) cases, for instance, it takes about 3 to 5 years for a PhD student to graduate from the university. After graduation, many students carry out their own research work, with much less or even no collaboration with their (PhD) supervisors. As a result, the attributes of collaboration networks will change. For example, collaboration frequency might decrease. On the other hand, it is difficult to determine when advisees will collaborate with their advisors since the collaboration pattern may vary from case to case.

Considering the above observations, we conduct experiments to examine the influence of edge attributes with different time lengths. Fig. 10 describes the effect of feature selection period with different durations in the edge autoencoder. Note that here we use only the edge autoencoder to avoid the influence of node attributes (i.e., academic age). We observe that the overall trend of all metrics increases with more investigated years in the fields of Physics, Computer Science, and Economics. Particularly, the edge autoencoder can achieve better performance if we adopt the collaboration information of first 7 years. It is noticeable that the performance differs from one field to another. The reason accounting for this phenomenon is the differences in collaboration patterns between advisees and advisors which exist among the disciplines.



(a) Accuracy (b) Precision

(c) Recall (d) F1-score

Fig. 10. Shifu2 performance with different time lengths of input data in edge autoencoder.

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

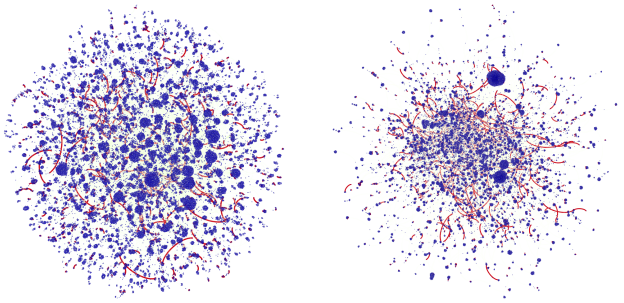
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

4.3 Application and Visualization

We apply Shifu2 onto the entire MAG dataset to generate academic genealogy automatically over all research fields. We know that some authors only publish one or two papers, and leave the academia at the early stage of their careers. In such cases, there is no long-term stable advisor-advisee relationship. How to select scholars with a stable advisor-advisee relationship is also a critical issue. Here we only apply Shifu2 to authors who meet the following criteria:

- The author has published at least 1 paper every 5 years;
- The author has published at least 10 papers in the entire MAG dataset;
- The author's publication career spans at least 10 years.

We calculate the personal attributes and collaboration attributes as the input of the node autoencoder and the edge autoencoder, respectively. By applying Shifu2, we generate a large-scale advisor-advisee relationship dataset. It is an enrichment containing not only advisor-advisee pairs but also their academic attributes. This dataset can be used in many applications, such as supervisor finding, academic performance assessment, and reviewer recommendation.



(a) Chemistry (b) Computer Science

Fig. 11. Collaboration networks in Chemistry and Computer Science.

For example, Fig. 11(a) and Fig. 11(b) show the collaboration networks in Chemistry and Computer Science, respectively. Nodes represent advisees and their collaborators including their advisors. We use red edges to represent the collaboration between advisees and their advisors. We can see that the collaboration patterns between advisees and advisors are quite different in these two areas. The collaborations in Computer Science are more concentrated than Chemistry, which verifies the disciplinary differences in scholars' collaboration patterns.

5 Conclusion

In this work, we extract advisor-advisee relationships based on scholars' publication records. We have proposed an efficient NRL model called Shifu2 to model and identify the advisor-advisee relationship. Specifically, we transform the large-scale network into the low-dimensional space and learn the representations for both nodes and edges based on autoencoder. Experiments upon real scientific collaboration networks demonstrate the effectiveness and stability of the proposed model. Furthermore, we have applied Shifu2 onto the entire MAG dataset to generate the large-scale academic genealogy automatically.

The following topics could be good choices for future work in this line of research.

(1) Both MAG and AFT have limited accuracy and completeness. For future work, more practical problems, for instance, how to acquire high-quality advisor-advisee pairs for training, could be considered. A potential solution is using the whole ProQuest[7] dissertation dataset to extract ground-truth advisor-advisee pairs.

(2) How to extend Shifu2 for identifying other types of relationships such as friendship in social networks?

(3) The use of the obtained benchmark dataset in various applications is yet to be explored.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (61872054) and the Fundamental Research Funds for the Central Universities (DUT19LAB23).

References

[1] J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing, and I. Lee, "Artificial intelligence in the 21st century," *IEEE Access*, vol. 6, no. 1, pp. 34 403–34 421, 2018.

[2] F. Xia, W. Wang, T. M. Bekele, and H. Liu, "Big scholarly data: A survey," *IEEE Transactions on Big Data*, vol. 3, no. 1, pp. 18–35, 2017.

[3] S. Fortunato, C. T. Bergstrom, K. Börner, J. A. Evans, D. Helbing, S. Milojević, A. M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi *et al.*, "Science of science," *Science*, vol. 359, no. 6379, p. eaao0185, 2018.

[4] M. E. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.

[5] W. Wang, Z. Cui, T. Gao, S. Yu, X. Kong, and F. Xia, "Is scientific collaboration sustainability predictable?" in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 853–854.

[6] S. Yu, F. Xia, K. Zhang, Z. Ning, J. Zhong, and C. Liu, "Team recognition in big scholarly data: Exploring collaboration intensity," in *The 15th IEEE International Conference on Big Data Intelligence and Computing (DataCom)*, 2017, pp. 925–932.

[7] G. T. Chao, P. Walz, and P. D. Gardner, "Formal and informal mentorships: A comparison on mentoring functions and contrast with nonmentored counterparts," *Personnel Psychology*, vol. 45, no. 3, pp. 619–636, 1992.

[8] J. Liu, T. Tang, X. Kong, A. Tolba, A.-M. Zafer, and F. Xia, "Understanding the advisor–advisee relationship via scholarly data analysis," *Scientometrics*, vol. 116, no. 1, pp. 1–20, 2018.

[9] R. D. Malmgren, J. M. Ottino, and L. A. N. Amaral, "The role of mentorship in protégé performance," *Nature*, vol. 465, no. 7298, p. 622, 2010.

[10] D. D. Beaver, "Reflections on scientific collaboration (and its study): past, present, and future," *Scientometrics*, vol. 52, no. 3, pp. 365–377, 2001.

[11] V. Larivière, Y. Gingras, C. R. Sugimoto, and A. Tsou, "Team size matters: Collaboration and scientific impact since 1900," *Journal of the Association for Information Science and Technology*, vol. 66, no. 7, pp. 1323–1332, 2015.

[12] R. K. Pan, A. M. Petersen, F. Pammolli, and S. Fortunato, "The memory of science: Inflation, myopia, and the knowledge network," *Journal of Informetrics*, vol. 12, no. 3, pp. 656–678, 2018.

[13] L. Liu, Y. Wang, R. Sinatra, C. L. Giles, C. Song, and D. Wang, "Hot streaks in artistic, cultural, and scientific careers," *Nature*, p. 1, 2018.

[14] Z. He, Z. Lei, and D. Wang, "Modeling citation dynamics of atypical articles," *Journal of the Association for Information Science and Technology*, 2018.

7. https://search.proquest.com/index

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

[15] W. Wang, J. Liu, F. Xia, I. King, and H. Tong, "Shifu: Deep learning based advisor-advisee relationship mining in scholarly big data," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 303–310.

[16] C. Wang, J. Han, Y. Jia, J. Tang, D. Zhang, Y. Yu, and J. Guo, "Mining advisor-advisee relationships from research publication networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 203–212.

[17] Z. Zhao, W. Liu, Y. Qian, L. Nie, Y. Yin, and Y. Zhang, "Identifying advisor-advisee relationships from co-author networks via a novel deep model," *Information Sciences*, vol. 466, pp. 258–269, 2018.

[18] C. P. Diehl, G. Namata, and L. Getoor, "Relationship identification for social network discovery," in *Proceedings of the 22nd National Conference on Artificial Intelligence*, vol. 1.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[20] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1067–1077.

[21] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 891–900.

[22] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proceedings of the 31st National Conference on Artificial Intelligence*, 2017, pp. 203–209.

[23] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 2111–2117.

[24] S. M. Kim, C. Paris, R. Power, and S. Wan, "Distinguishing individuals from organisations on twitter," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 805–806.

[25] C. Tu, H. Liu, Z. Liu, and M. Sun, "Cane: Context-aware network embedding for relation modeling," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2017, pp. 1722–1731.

[26] C. Tu, Z. Zhang, Z. Liu, and M. Sun, "Transnet: Translation-based network representation learning for social relation extraction," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2017.

[27] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[29] P. V. Marsden, "Homogeneity in confiding relations," *Social Networks*, vol. 10, no. 1, pp. 57–76, 1988.

[30] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] T. Wu, Y. Chen, and J. Han, "Re-examination of interestingness measures in pattern mining: a unified framework," *Data Mining and Knowledge Discovery*, vol. 21, no. 3, pp. 371–397, 2010.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] F. E. Harrell, "Ordinal logistic regression," in *Regression Modeling Strategies*, 2015, pp. 311–325.

[35] H. Iglič, P. Doreian, L. Kronegger, and A. Ferligoj, "With whom do researchers collaborate and why?" *Scientometrics*, vol. 112, no. 1, pp. 153–174, 2017.

[36] F. G. Montoya, A. Alcayde, R. Baños, and F. Manzano-Agugliaro, "A fast method for identifying worldwide scientific collaborations using the scopus database," *Telematics and Informatics*, vol. 35, no. 1, pp. 168–185, 2018.

[37] R. Sinatra, D. Wang, P. Deville, C. Song, and A.-L. Barabási, "Quantifying the evolution of individual scientific impact," *Science*, vol. 354, no. 6312, p. aaf5239, 2016.

[38] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.

Jiaying Liu received the BSc degree in software engineering from Dalian University of Technology, China, in 2016. She is currently working toward the Ph.D. degree in the School of Software, Dalian University of Technology, China. Her research interests include data science, big scholarly data, and social network analysis.

Feng Xia (M'07-SM'12) received the BSc and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently a Full Professor in School of Software, Dalian University of Technology, China. Dr. Xia has published 2 books and over 200 scientific papers in international journals and conferences. His research interests include data science, knowledge engineering, and systems engineering. He is a Senior Member of IEEE and ACM.

Lei Wang received the BSc degree in software engineering from Dalian University of Technology, China, in 2018. He is currently working toward the master's degree in the School of Software, Dalian University of Technology, China. His research interests include big scholarly data and network science.

Bo Xu received the BSc and PhD degrees from the Dalian University of Technology, China, in 2007 and 2014, respectively. She is currently a lecturer in School of Software at the Dalian University of Technology. Her current research interests include social computing, data mining, information retrieval, and natural language processing.

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2946825, IEEE Transactions on Knowledge and Data Engineering

Xiangjie Kong (M'13-SM'17) received the B-Sc and PhD degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor in School of Software, Dalian University of Technology, China. His research interests include computational social science, data science, and city science. He is a Senior Member of IEEE and a Member of ACM.

Hanghang Tong is currently an Assistant Professor at School of Computing, Informatics, and Decision Systems Engineering (CIDSE), Arizona State University since August 2014. He received his M.Sc and Ph.D. degree from Carnegie Mellon University in 2008 and 2009, both majored in Machine Learning. His research interest is in large scale data mining for graphs and multimedia.

Irwin King (F'18) received the B.Sc. degree in engineering and applied science from the California Institute of Technology, Pasadena, CA, USA, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, CA. He is currently the Associate Dean (Education) of the Faculty of Engineering, and a Professor at The Chinese University of Hong Kong. His research interests include machine learning, social computing, big data, Web intelligence, data mining, and multimedia information processing. He is a Fellow of IEEE.

1041-4347 (c) 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.